

Competidor(a): \_\_\_\_\_

Número de inscrição: \_\_\_\_\_ – \_\_\_\_\_ (opcional)

*Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (01 de setembro de 2023).*



Olimpíada Brasileira de Informática  
Competição Feminina - OBI2023

Caderno de Tarefas  
Modalidade Programação • Nível Júnior • Fase Única

01 de setembro de 2023

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



# Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 8 páginas (não contando a folha de rosto), numeradas de 1 a 8. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Você pode submeter até 50 soluções para cada tarefa. A pontuação total de cada tarefa é a melhor pontuação entre todas as submissões. Se a tarefa tem sub-tarefas, para cada sub-tarefa é considerada a melhor pontuação entre todas as submissões.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
  - em C: *scanf*, *getchar*, *printf*, *putchar*;
  - em C++: as mesmas de C ou os objetos *cout* e *cin*.
  - em Java: qualquer classe ou função padrão, como por exemplo *Scanner*, *BufferedReader*, *BufferedWriter* e *System.out.println*
  - em Python: *read*, *readline*, *readlines*, *input*, *print*, *write*
  - em Javascript: *scanf*, *printf*
- Procure resolver a tarefa de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

# Conversa dos Pinguins

Nome do arquivo: `pinguins.c`, `pinguins.cpp`, `pinguins.java`, `pinguins.js` ou `pinguins.py`

Ally e Billy são pinguins muito tecnológicos e adoram jogar jogos, assistir vídeos e conversar pela Internet. Mesmo morando longe um do outro, eles trocam mensagens todos os dias utilizando o Icebook, que é a rede social mais popular entre os pinguins.

Um dos seus assuntos favoritos é a temperatura do local onde cada um está. Ally teve a ideia de até mesmo transformar isso em um jogo: todos os dias, exatamente ao meio-dia, os dois pinguins utilizam seus Termômetros Oficiais de Pinguim (TOPs) para medir a temperatura. TOPs utilizam uma escala de temperatura especial desenvolvida pelos mais brilhantes pinguins cientistas e são usados por todos os pinguins. O pinguim que estiver no lugar mais frio ganha um ponto.

Passeando recentemente pela Antártida, os dois conheceram exploradores de diversos países e ficaram surpresos ao descobrir que os seres humanos também inventaram escalas de temperatura. Fascinados com a novidade, Ally e Billy perguntaram para os exploradores se aceitariam fazer uma troca de presentes: TOPs por termômetros humanos. Entre os termômetros que ganharam, eles notaram que alguns utilizam uma escala chamada Celsius e outros utilizam uma escala chamada Fahrenheit. Os exploradores explicaram que as duas escalas se relacionam utilizando a seguinte fórmula, onde  $C$  é a temperatura na escala Celsius e  $F$  é a temperatura na escala Fahrenheit:

$$C = (F - 32) \times 5/9$$

Billy então sugeriu que fizessem uma modificação em seu jogo: primeiro, cada umalaria a escala do termômetro humano que utilizaram para fazer a medição. Em seguida, cada umalaria a temperatura do local onde cada um está. Utilizando a fórmula da explicação dos exploradores, eles descobririam então quem ganhou o ponto.

A sua tarefa é fazer um programa para ajudar Ally e Billy a descobrir quem vai ganhar o ponto.

## Entrada

A primeira linha da entrada contém dois caracteres,  $E_A$  e  $E_B$ , representando a escala de temperatura utilizada por Ally e Billy, respectivamente. Os possíveis valores são “C” (sem aspas) para a escala Celsius e “F” (sem aspas) para a escala Fahrenheit.

A segunda linha da entrada contém dois números inteiros,  $T_A$  e  $T_B$ , representando as temperaturas medidas por Ally e Billy, respectivamente.

## Saída

Sua saída deve conter o caractere “A” (sem aspas) se Ally estiver no local mais frio ou “B” (sem aspas) se Billy estiver no local mais frio. É garantido que a temperatura nunca será igual nos dois locais.

## Restrições

- $-130 \leq T_A, T_B \leq 68$
- $E_A, E_B \in \{C, F\}$

## Informações sobre a pontuação

- Em um conjunto de casos de teste somando 7 pontos, tanto Ally quanto Billy utilizam a escala Celsius.

- Em um conjunto de casos de teste somando 7 pontos, tanto Ally quanto Billy utilizam a escala Fahrenheit.
- Em um conjunto de casos de teste somando 14 pontos, Ally sempre utiliza a escala Celsius e Billy sempre utiliza a escala Fahrenheit.
- Em um conjunto de casos de teste somando 14 pontos, Ally sempre utiliza a escala Fahrenheit e Billy sempre utiliza a escala Celsius.
- Em um conjunto de casos de teste somando 58 pontos, nenhuma restrição adicional.

## Exemplos

|  |                                |
|--|--------------------------------|
| <b>Exemplo de entrada 1</b><br>C C<br>-10 10 | <b>Exemplo de saída 1</b><br>A |
| <b>Exemplo de entrada 2</b><br>C F<br>-10 5  | <b>Exemplo de saída 2</b><br>B |
| <b>Exemplo de entrada 3</b><br>F C<br>0 0    | <b>Exemplo de saída 3</b><br>A |

# Estante de Livros

Nome do arquivo: `estante.c`, `estante.cpp`, `estante.java`, `estante.js` ou `estante.py`

A princesa Jujuba encontrou 3 pilhas em uma sala secreta de seu castelo. A primeira pilha tem  $X$  livros, a segunda pilha tem  $Y$  livros e a terceira pilha tem  $Z$  livros. Jujuba quer colocar seus livros na sua estante, que tem  $N$  prateleiras.

Cada prateleira pode conter infinitos livros, mas como ela é organizada, é preciso que todas as prateleiras tenham a mesma quantidade de livros. Como nem sempre isso é possível, Jujuba quer saber qual a menor quantidade de livros  $L$  que terão de ficar fora da sua estante.

Um exemplo seria as pilhas terem 2, 5 e 9 livros e sua estante ter 3 prateleiras. Jujuba poderia colocar 1 livro em cada prateleira, deixando 13 livros de fora. Mas a melhor resposta seria Jujuba colocar 5 livros em cada prateleira, deixando apenas 1 livro de fora.

Você consegue ajudar a princesa Jujuba?

## Entrada

A única linha de entrada contém quatro inteiros  $X$ ,  $Y$ ,  $Z$  e  $N$ , descritos no enunciado.

## Saída

Seu programa deve produzir uma única linha contendo um único inteiro  $L$ .

## Restrições

- $1 \leq X, Y, Z, N \leq 10^{18}$

## Informações sobre a pontuação

- $X, Y, Z, N \leq 10^6$  (73 pontos)
- Sem mais restrições (27 pontos)

## Exemplos

|  |                                |
|--|--------------------------------|
| <b>Exemplo de entrada 1</b><br>2 5 9 3 | <b>Exemplo de saída 1</b><br>1 |
| <b>Exemplo de entrada 2</b><br>1 2 3 4 | <b>Exemplo de saída 2</b><br>2 |

# Suco Radioativo

Nome do arquivo: `suco.c`, `suco.cpp`, `suco.java`, `suco.js` ou `suco.py`

Letícia estava com muita sede e decidiu fazer um suco de abacaxi, seu favorito, para que ela pudesse tomar. Como sua última experiência não foi muito boa, ela fez um suco estragado, ela pede a você que determine quantos copos de suco ela pode tomar. Ela só irá tomar os copos que não estão contaminados.

Um suco é dito como contaminado se e somente se ele não é de abacaxi com hortelã e possui “pedacinhos” nele. Para cada copo é dado dois valores “ $A$ ” e “ $B$ ”. “ $A$ ” indica se o suco é de abacaxi com hortelã ou não e “ $B$ ” indica se o suco tem pedaços nele. Sendo assim, determine quantos copos de suco ela poderá tomar.

## Entrada

A primeira linha contém um inteiro  $N$  que representa a quantidade de copos de suco disponíveis. As próximas  $N$  linhas contém cada uma dois inteiros  $A$  e  $B$ , respectivamente. O valor  $A$  indicará se o  $i^{th}$  suco é de abacaxi ou não e o valor  $B$  indicará se o suco possui pedaços nele.

$A$  irá valer 1 caso o suco seja de abacaxi com hortelã e 0 caso contrário.  $B$  irá valer 1 caso o suco tenha pedaços e 0 caso contrário.

## Saída

Imprima um inteiro representando a quantidade de copos de suco que ela pode tomar, ou seja que não estão estragados.

## Restrições:

- $1 \leq N \leq 5 \cdot 10^5$
- $0 \leq A \leq 1$
- $0 \leq B \leq 1$

## Informações sobre a pontuação:

- $N = 1$  (16 pontos)
- Sem mais restrições (84 pontos)

## Exemplos

| Entrada                              | Saída |
|--------------------------------------|-------|
| 5<br>1 1<br>0 0<br>1 0<br>0 1<br>0 1 | 3     |

| Entrada                | Saida |
|------------------------|-------|
| 3<br>0 0<br>0 1<br>1 0 | 2     |

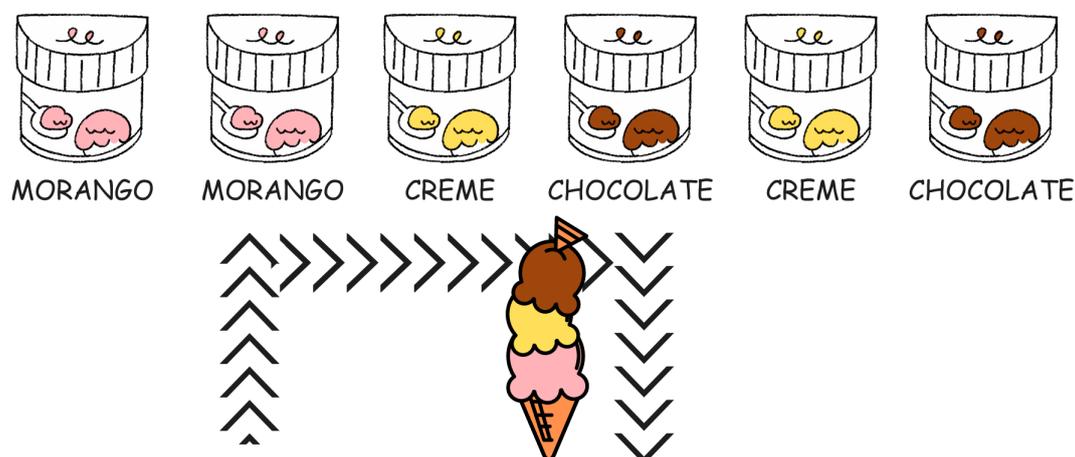
# A Grande Casquinha

Nome do arquivo: `casquinha.c`, `casquinha.cpp`, `casquinha.java`, `casquinha.js` ou `casquinha.py`

Samyra gosta muito de sorvete e quer montar uma casquinha gigante com o máximo de bolas possíveis! Porém, ela é uma garota muito enjoada, então se ela já pegou uma bola com certo sabor de sorvete ela se recusará a pegar outra bola com o mesmo sabor.

Como a sorveteria da cidade anda muito cheia, eles criaram uma regra:

Você pode entrar no balcão da sorveteria em qualquer posição e andar para a direita pegando uma bola de cada pote de sorvete pelo qual você passar, mas no momento que você recusar pegar o sorvete de um pote você tem que finalizar sua casquinha e se dirigir ao caixa. (Para evitar que pessoas indecisas fiquem ocupando o balcão).



## Entrada

A primeira linha da entrada contém um inteiro  $N$ , o número de potes de sorvete no balcão. A segunda linha contém um inteiro  $S$ , número máximo de sabores distintos de sorvete. As próximas  $N$  linhas contém cada uma um inteiro  $X_i$ , o sabor do  $i$ -ésimo sorvete.

## Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o número de bolas na maior casquinha que Samyra pode montar.

## Restrições

- $1 \leq N \leq 120\,000$
- $1 \leq S \leq 10\,000\,000$
- $1 \leq X_i \leq S, 1 \leq i \leq N$

### Informações sobre a pontuação

- Em um conjunto de teste valendo 11 pontos,  $S = 2$ .
- Em um conjunto de teste valendo 12 pontos,  $S = 4$ .
- Em um conjunto de teste valendo 14 pontos,  $1 \leq N \leq 200$ .
- Em um conjunto de teste valendo 19 pontos,  $1 \leq S \leq 100$ .
- Em um conjunto de teste valendo 17 pontos,  $1 \leq N \leq 5\,000$ .
- Em um conjunto de teste valendo 27 pontos, sem restrições adicionais.

### Exemplos

| Exemplo de entrada 1  | Exemplo de saída 1 |
|-----------------------|--------------------|
| 3<br>2<br>1<br>2<br>1 | 2                  |

| Exemplo de entrada 2                 | Exemplo de saída 2 |
|--------------------------------------|--------------------|
| 6<br>3<br>1<br>1<br>2<br>3<br>2<br>3 | 3                  |